

#31/Dec. Supp.



PATENT

Attorney Docket No.: 23452-012

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Douglas Walter CONMY, et al.

SERIAL NUMBER : 09/100,133

EXAMINER : S.M. Meinecke Diaz

FILING DATE : June 19, 1998

ART UNIT : 2163

FOR : ELECTRONIC CALENDAR WITH GROUP SCHEDULING  
AND STORAGE OF USER AND RESOURCE PROFILESAssistant Commissioner for Patents  
Washington, D.C. 20231**RECEIVED**

APR 24 2003

SUPPLEMENTAL DECLARATION OF ROBERT SLAPIKOFF**GROUP 3600**

Sir:

I, Robert Slapikoff hereby declare as follows:

1. I am an inventor of the invention claimed in the above-identified patent application.
2. I have reviewed the Office Action of August 14, 2002, for the subject application, and I have also reviewed the current status of the claims.
3. The publication by Sybex titled, "The ABCs of Outlook 97", dated March 17, 1997, is alleged to describe subject matter similar to that disclosed in the subject patent application.
4. The publication by PC World Magazine titled "The Future is Bright for Microsoft Outlook 97" dated November 1, 1996, is also alleged to describe subject matter similar to that disclosed in the subject patent application.
5. "Running Microsoft Windows" by Microsoft Press - dated June 25, 1997, is also alleged to describe subject matter similar to that disclosed in the above-identified patent application.

Douglas Walter CONMY, et al.  
09/100,133

6. Prior to November 1, 1996, Douglas Walter Conmy, Steven Robert Beckhardt, John Banks-Binici and I collectively completed the invention claimed in the above-identified patent application.
7. Prior to November 1, 1996, Douglas Walter Conmy, Steven Robert Beckhardt, John Banks-Binici and I conceived of an reduced the invention to practice.
8. Since the claimed invention is incorporated in Lotus Notes version 4.50, operable on a personal computer, for example, in communication with a computer network, which was released on December 4, 1996, the claimed invention was completed and reduced to practice well before this date, as such software releases usually take, at a minimum, more than a year to prepare.
9. As evidence of the invention being completed and reduced to practice prior to November 1, 1996, a listing of the source code files which provides the features of the claimed invention is set out on Exhibit A. This information was obtained from a software archive from the assignee's engineering department.
10. The source code file nsf/schedule.c, corresponds to a process by which one or more invitee profiles for one or more potential invitees of the system are stored. Each profile includes information regarding available and unavailable times for that user. Specifically, the process enables a server to collect a user's available time known as "working hours" (i.e., free-time), as well as time that the user has blocked off (i.e., busy-time) into a database/file for optimized retrieval. The information may be distributed across multiple servers as is made available to requesting individuals. This process also monitors a user's accepted and scheduled appointments and reflects them in the database.
11. The source code files nsf/schedule.c, client/clschret.c and server/scschret.c correspond to a process by which a request for allocation of a time interval for the one or more potential invitees is received. A Lotus Notes client or a Lotus Domino server may obtain the availability of one or more users by calling the application-program-interface (API)

Douglas Walter CONMY, et al.  
09/100,133

functions SchRetrieve (retrieve a schedule) or SchSrvRetrieve contained in the above-noted source code files.

12. The function SchRetrieve is a function which synchronously retrieves a local or remote schedule by asking the caller's home server for the schedule. The only time that local busy time is used is when the client is in a "Disconnected" mode, which is specified through a location document. Otherwise, the API will route all lookup requests to the users home server for processing. Exhibit B includes further documentation for the SchRetrieve function.
13. SchSrvRetrieve is a function which, upon being used on a server, synchronously retrieves local or remote schedules from the proper fanout servers. Upon the SchSrvRetrieve being used on a client, only locally available information is retrieved in busytime.nsf. Exhibit C includes further documentation for the SchSrvRetrieve function.
14. The source-code files nsf/schedule.c, nsf/schcntnr.c, nsf/schobj.c and nsf/schods.c correspond to a process by which invitee profiles for the one or more potential invitees are gathered. This is initiated in response to the request for the allocation of a time interval for the one or more invitees, and the information is obtained from the stored invitee profile data. The data is represented in memory by schcntnr.c and schobj.c., in a SchContainer. Access to the data within the container is made possible via the API functions SchContainer\_GetFirstSchedule and SchContainer\_GetNextSchedule.
15. SchContainer\_GetFirstSchedule. This function is used to get a handle to the first schedule object in a container (see also Exhibit D).
16. SchContainer\_GetNextSchedule. This API function is used to get a handle to the a next schedule object in a container (see also Exhibit E).

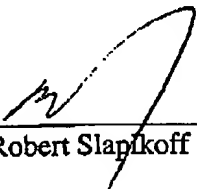
Douglas Walter CONMY, et al.  
09/100,133

17. Source code files `schui/bsysugg.cpp` and `misc/timelist.c` correspond to a process for determining whether potential invitees are available during the requested time interval. This process specifically determines the availability of one or more users by first requesting their freetime and busytime (see part 10 above). Once this information is obtained, the source code provides further processing of this data via a time weighting algorithm designed to find a period of time where all users are available, and, if not possible, a time where most of the users are available.
18. Source code `schui/bsylist.cpp`, `schui/bsylistw.cpp`, `schui/freetime.cpp` and `wmisc/drawtime.cpp` enables a process by which results of the determination (see part 17 above) are displayed by permitting a user to select from at least three results viewing options including a viewing option displaying the one or more potential invitees that are available, a viewing option displaying the one or more potential invitees that are not available and a viewing option displaying the one or more potential invitees whose schedule could not be found, and then displaying the results according to the option selected.
19. Using an email program, in the present invention, an email is sent from an event coordinator which includes event information for scheduling an event where the event information specifying a list of invitees, a date, a start time, and an end time and/or a duration which determines the end time.
20. While the majority of the files were last modified prior to November 1, 1996, several of them were changed either on 11/12/96 and 11/15/96 (3 files). To that end, at that point in our development cycle, given that the final release was well underway (released December 4, 1996), the changes made to those files were only made to fix software bugs, and did not add or extend any of the existing functionality or capabilities of the invention. This can be further evidenced by the following development comment files regarding changes made to those files around that time as shown in Exhibit F.

Douglas Walter CONMY, et al.  
09/100,133

21. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under Section 1001 of Title 18 of the United States Code and that such willful and false statements may jeopardize the validity of the application or any patent issued thereon.

Date 1/10/2003

  
\_\_\_\_\_  
Robert Slapikoff

Douglas Walter CONMY, et al.  
09/100,133

## EXHIBIT A

m:\v4x\v4.5\client\clschret.c	20121	10-23-1996 10:14:56
...		
m:\v4x\v4.5\misc\timelist.c	70828	11-15-1996 11:27:34
...		
m:\v4x\v4.5\nsf\schentnr.c	55805	11-12-1996 16:42:50
m:\v4x\v4.5\nsf\schedule.c	171683	11-12-1996 16:37:06
m:\v4x\v4.5\nsf\schobj.c	41280	10-24-1996 15:12:18
m:\v4x\v4.5\nsf\schods.c	27287	10-12-1996 14:07:42
m:\v4x\v4.5\nsf\schrqst.c	23671	10-22-1996 13:51:10
...		
m:\v4x\v4.5\schedule\main.c	105705	10-31-1996 7:13:08
...		
m:\v4x\v4.5\schui\bsydata.cpp	21145	10-31-1996 11:52:30
m:\v4x\v4.5\schui\bsyinvit.cpp	17617	10-29-1996 11:55:52
m:\v4x\v4.5\schui\bsylist.cpp	16207	10-29-1996 13:05:06
m:\v4x\v4.5\schui\bsylist.hpp	7052	10-29-1996 13:06:02
m:\v4x\v4.5\schui\bsylistw.cpp	9376	10-15-1996 17:12:02
m:\v4x\v4.5\schui\bsysugg.cpp	13703	10-22-1996 15:09:44
m:\v4x\v4.5\schui\freetime.cpp	28819	10-22-1996 18:41:48
...		
m:\v4x\v4.5\server\svschret.c	13242	10-14-1996 18:10:38
...		
m:\v4x\v4.5\wnisc\drawtime.c	8266	9-16-1996 13:29:08

Douglas Walter CONMY, et al.  
09/101,133

## EXHIBIT B

Initial Release 4.5

Function : Calendaring and Scheduling

SchRetrieve - Retrieve a schedule.

---

#include <schedule.h>

```
STATUS LNPUBLIC SchRetrieve(  
    UNID FAR *pApptUnid,  
    TIMEDATE FAR *pApptOrigDate,  
    DWORD dwOptions,  
    TIMEDATE_PAIR FAR *pInterval,  
    LIST FAR *pNames,  
    HCONTNR FAR *rethCntnr,  
    void FAR *MustBeNull1,  
    void FAR *MustBeNull2,  
    void FAR* FAR *MustBeNull3);
```

Description :

Synchronously retrieves a local or remote schedule by asking the caller's home server for the schedule.

The ONLY time that local busy time is used is when the client is in the Disconnected mode which is specified through the location document. Otherwise, the API will route ALL lookup requests to the users home server for processing.

Parameters :

Input :

pApptUnid - Ignore this UNID in computations.

pApptOrigDate - Reserved. Must be set to NULL.

dwOptions - Option flags:

SCHRQST\_COMPOSITE return composite schedule

SCHRQST\_EACHPERSON return each person's schedule

SCHRQST\_LOCAL do only local lookup

SCHRQST\_FORCEREMOTE force remote even if you are using workstation based email

pInterval - Pointer to a TIMEDATE\_PAIR structure that specifies the range over which the free time search should be performed. In typical scheduling applications, this might be a range of 1 day or 5 days.

pNames - Pointer to a list of fully distinguished names whose schedule should be searched. This list is in TEXT\_LIST format without the datatype

Douglas Walter CONMY, et al.  
09/100,133

work. This list can be conveniently built with the textlist package.

MustBeNull1 - This parameter must be NULL.

MustBeNull2 - This parameter must be NULL.

MustBeNull3 - This parameter must be NULL.

Output :

(routine) - NOERROR - Successfully retrieved a schedule.

ERR\_XXX - There are many possible errors. It is best to use the code in a call to OSLoadString and display/log the error for the user as your default error handling.

rethCntnr - Handle of schedule container results are returned in. If \*rethCntnr is NULLHANDLE, then the container will be allocated by this call. If it is not NULLHANDLE then the caller has allocated it and is responsible for freeing it on all errors.

See Sample Program :

MISC\SCHEDULE

See Also :

SchContainer\_Free  
SchSrvRetrieve

SCHRQST\_XXX



Douglas Walter CONMY, et al.  
09/100,133

## EXHIBIT C

Initial Release 4.5

Function : Calendaring and Scheduling

SchSrvRetrieve - Retrieve a schedule.

---

```
#include <schedule.h>
```

```
STATUS LNPUBLIC SchSrvRetrieve(
    LIST FAR *pClientNames,
    UNID FAR *pApptUnid,
    TIMEDATE FAR *pApptOrigDate,
    DWORD dwOptions,
    TIMEDATE_PAIR FAR *pInterval,
    LIST FAR *pNames,
    HCONTNR FAR *rethCntnr);
```

### Description :

When this call is made on a server, it synchronously retrieves local or remote schedules from the proper fanout servers. When this call is made on a client, it retrieves only locally available information in busytime.nsf

This container must be deallocated by the caller using SchContainer\_Free.

### Parameters :

#### Input :

pClientNames - List of users making query.

pApptUnid - Ignore this UNID in computations.

pApptOrigDate - This is the original start date of the appointment to ignore. This is only here for Organizer 2.x compatibility.

dwOptions - Option flags:

SCHRQST_COMPOSITE	default if dwOptions is 0
SCHRQST_EACHPERSON	return each person's schedule
SCHRQST_LOCAL	only look up schedules locally

pInterval - Pointer to a TIMEDATE\_PAIR structure that specifies the range over which the free time search should be performed. In typical scheduling applications this might be a range of 1 day or 5 days.

pNames - Pointer to a list of distinguished names whose schedule should be searched. This list is in TEXT\_LIST format without the datatype word.

Douglas Walter CONMY, et al.  
09/100,133

This list can be build with the textlist package (e.g. ListAllocate)

Output :

(routine) - NOERROR - Successfully retrieved a schedule.

ERR\_XXX - There are many possible errors. It is best to use the code in a call to OSLoadString and display/log the error for the user as your default error handling.

rethCntnr - Handle of the container that results are returned in.

See Also :

SchContainer\_Free  
SCHRQST\_XXX  
SchRetrieve

Douglas Walter CONMY, et al.  
09/100,133

## EXHIBIT D

Initial Release 4.5

Function : Calendaring and Scheduling

SchContainer\_GetFirstSchedule - Get first schedule object in a container.

---

```
#include <schedule.h>
```

```
STATUS LNPUBLIC SchContainer_GetFirstSchedule(  
    HCNTNR hCntnr,  
    HSCHEDULE FAR *rethObj,  
    SCHEDULE FAR * FAR *retpSchedule);
```

Description :

This function is used to get a handle to the first schedule object in a container.

Parameters :

Input :

hCntnr - The container handle.

Output :

(routine) - NOERROR - Successfully got first schedule.

ERR\_XXX - There are many possible errors. It is best to use the code in a call to OSLoadString and display/log the error for the user as your default error handling.

rethObj - Points to where we return the handle to the schedule.

retpSchedule - Points to where we return a pointer to the actual data.

See Sample Program :

MISC\SCHEDULE

See Also :

SchContainer\_GetNextSchedule

Douglas Walter CONMY, et al.  
09/100,133

## EXHIBIT E

Initial Release 4.5

Data Type : Calendaring and Scheduling

SCHEDULE - Data structure for a schedule.

---

```
#include <schedule.h>
```

Definition :

```
typedef struct {
    DWORD    reserved[8];
    DBID      dbReplicaID; /* Users mail file replica ID */
    TIMEDATE_PAIR Interval; /* events etc. are in this
                           interval */
    DWORD     dwErrGateway; /* gateway error retrieving this
                           schedule */
    STATUS     error; /* error retrieving this
                     schedule */
    WORD       wReserved; /* unused at this time */
    WORD       wOwnerNameSize; /* size of owner name
                           (includes term.) */
} /* followed by owner name */
SCHEDULE;
```

Description :

Data structure for a schedule.

See Sample Program :

MISC\SCHEDULE

See Also :

SchContainer\_GetFirstSchedule  
SchContainer\_GetNextSchedule  
SchContainer\_FindSchedule

Douglas Walter CONMY, et al.  
09/100,133

## EXHIBIT F

From kit\dwc.txt, comments from Douglas Comny:

Build 145.1

DWC 11/02(96) misc: Fixed incorrect profile display when the date/time crosses DST. [timelist.c] (#DCOY3B2ML6)

From kit\jeb.txt, comments from John Banks-Binici:

Build 145

JEB 9/19/96 nsf (145) check hCntr for NULL before calling  
OSLock. Fix client to handle ctrl-break (#MMOO39M2EL)  
[schobj.c][schods.c] [schcntnr.c] [client\clschret.c]

JEB 9/19/96 nsf (145) don't infinitely loop on Calendar domain if it  
is same as current domain (#MMOO39M27U) (schedule.c)

JEB 10/4/96 bcase (145) allow NEXT button to work right when  
replicating free time info (#DCOY3A2SZK)  
[nsfschedule.c][server\svschret.c][bcase\bcdsched.c]